Decomposition approaches for two-stage robust binary optimization

Ayşe N Arslan, Boris Detienne

Decomposition approaches for two-stage robust binary optimization

Ayşe-Nur Arslan [1], Boris Detienne [2]

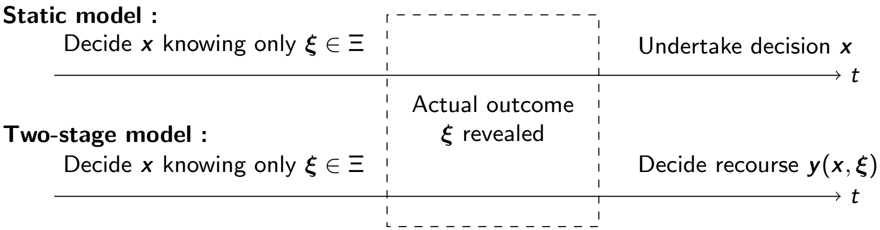
[1]Génie Mathématique, INSA de Rennes
Institut de Recherche Mathématique de Rennes - UMR CNRS 6625

[2]Institut de Mathématiques de Bordeaux, UMR CNRS 5251, Université de Bordeaux
RealOpt, Inria Bordeaux Sud–Ouest

Outline

1. Introduction
2. Theoretical development
3. Application to robust capital budgeting
4. Numerical results
5. Conclusion

Static vs Two-stage robust optimization



Mixed integer linear robust optimization with recourse

$$\min_{\boldsymbol{x}\in\mathcal{X}} \ \boldsymbol{c}^\top \boldsymbol{x} + \max_{\boldsymbol{\xi}\in\Xi} \ \min_{\boldsymbol{y}\in\mathcal{Y}(\boldsymbol{x},\xi)} \ \boldsymbol{\xi}^\top \boldsymbol{Q}\boldsymbol{y}$$

LP (MIP) model [semi-infinite]

$$\begin{aligned}
(CR):\ \min\ & \boldsymbol{c}^\top \boldsymbol{x} + t \\
\text{s.t.}\ & \boldsymbol{x} \in \mathcal{X} \\
& t \geq \boldsymbol{q}(\boldsymbol{\xi})\boldsymbol{y}(\boldsymbol{\xi}) && \forall \boldsymbol{\xi} \in \Xi \\
& \boldsymbol{T}(\boldsymbol{\xi})\boldsymbol{x} + \boldsymbol{W}(\boldsymbol{\xi})\boldsymbol{y}(\boldsymbol{\xi}) \leq \boldsymbol{h}(\boldsymbol{\xi}) && \forall \boldsymbol{\xi} \in \Xi \\
& \boldsymbol{y}(\boldsymbol{\xi}) \in \mathcal{Y} && \forall \boldsymbol{\xi} \in \Xi
\end{aligned}$$

- $\boldsymbol{x}$: first-stage decisions
- $\boldsymbol{t}$: cost of recourse
- $\boldsymbol{y}(\boldsymbol{\xi})$: recourse decisions
- $\boldsymbol{q}(\boldsymbol{\xi}), \boldsymbol{T}(\boldsymbol{\xi}), \boldsymbol{W}(\boldsymbol{\xi}), \boldsymbol{h}(\boldsymbol{\xi})$: uncertainty revealed after first-stage decisions are taken

Literature review

1. Exact approaches: Often based on dual information or using a facial description of the recourse polyhedron:
   - (i) Constraint generation: Atamturk and Zeng (2007), Thiele et al. (2009), Bertsimas et al. (2013), Jiang et al. (2014), Zhen et al. (2018)
   - (ii) Constraint-and-Column generation: Ayoub and Poss (2016), Zhao and Zeng (2012), Zeng and Zhao (2013)
   - (iii) Convexification-based : Kämmerling and Kurtz (2019)
2. Approximate approaches:
   - (i) Decision rules: Recourse decisions are restricted to be functions of uncertainty: Ben-Tal et al. (2004), Chen and Zhang (2009), Goh and Sim (2010), Kuhn et al. (2011), Vayanos et al. (2011), Georghiou et al. (2015), Bertsimas and Dunning (2016), Bertsimas and Georghiou (2015,2018), Gorissen et al. (2015), Postek and den Hertog (2016)
   - (ii) K-adaptability: K recourse decisions are selected at the first stage, optimization is done over them in the second stage: Hanasusanto et al. (2015), Subramanyam et al. (2017), Buchheim and Kurtz (2017)

Our contribution

In this presentation:

- We study a class of two-stage mixed binary robust optimization problems with objective uncertainty
- We present an exact solution method for these problems
- Our method uses convexification and dualization as a tool for obtaining a deterministic equivalent formulation
- It uses branch-and-price algorithm to solve the resulting model
- We numerically compare our algorithm with the approximate K-adaptability approach

Outline

1. Introduction
2. Theoretical development
3. Application to robust capital budgeting
4. Numerical results
5. Conclusion

Two-stage robust binary optimization problem with objective uncertainty

$$\min_{\boldsymbol{x}\in\mathcal{X}} \quad \boldsymbol{c}^\top \boldsymbol{x} + \max_{\boldsymbol{\xi}\in\Xi} \min_{\boldsymbol{y}\in\mathcal{Y}(\boldsymbol{x})} \quad \boldsymbol{\xi}^\top \boldsymbol{Q}\boldsymbol{y} \tag{1}$$

1. Bounded mixed binary sets $\mathcal{X} \subseteq \{0,1\}^{N_1} \times \mathbb{R}_+^{N_2}$, $\mathcal{Y} \subseteq \{0,1\}^{M_1} \times \mathbb{R}_+^{M_2}$
2. Bounded polyhedral set $\Xi \subseteq \mathbb{R}^Q$, affects only the objective function
3. $\mathcal{Y}(\boldsymbol{x}) = \{\boldsymbol{y} \in \mathcal{Y} | \boldsymbol{H}\boldsymbol{y}_1 \leq \boldsymbol{d} - \boldsymbol{T}\boldsymbol{x}_1\}$

Notation: $\boldsymbol{x} = \begin{pmatrix} \boldsymbol{x}_1 \\ x_* \end{pmatrix}$ and $\boldsymbol{y} = \begin{pmatrix} \boldsymbol{y}_1 \\ y_* \end{pmatrix}$ with $\boldsymbol{x}_1 \in \{0,1\}^{L_1}, \boldsymbol{y}_1 \in \{0,1\}^{L_2}$.

Reduction to a static robust problem

Proposition

Problem (1) *is equivalent to*

$$\min_{\boldsymbol{x}\in\mathcal{X},\boldsymbol{y}\in\mathrm{conv}(\mathcal{Y}(\boldsymbol{x}))} \quad \boldsymbol{c}^\top \boldsymbol{x} + \max_{\boldsymbol{\xi}\in\Xi} \quad \boldsymbol{\xi}^\top \boldsymbol{Q}\boldsymbol{y} \tag{2}$$

Proof.

For given $\boldsymbol{x}\in\mathcal{X}$, and $\boldsymbol{\xi}\in\Xi$,

$$\min_{\boldsymbol{y}\in\mathcal{Y}(\boldsymbol{x})} \quad \boldsymbol{\xi}^\top \boldsymbol{Q}\boldsymbol{y} = \min_{\boldsymbol{y}\in\mathrm{conv}(\mathcal{Y}(\boldsymbol{x}))} \quad \boldsymbol{\xi}^\top \boldsymbol{Q}\boldsymbol{y}.$$

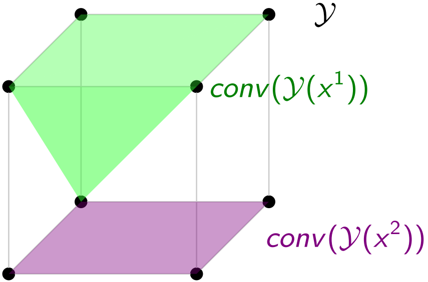
Apply the minimax theorem on $\max_{\boldsymbol{\xi}\in\Xi}\min_{\boldsymbol{y}\in\mathrm{conv}(\mathcal{Y}(\boldsymbol{x}))} \quad \boldsymbol{\xi}^\top \boldsymbol{Q}\boldsymbol{y}$

1. $\boldsymbol{\xi}^\top \boldsymbol{Q}\boldsymbol{y}$ is convex in $\boldsymbol{y}$
2. $\boldsymbol{\xi}^\top \boldsymbol{Q}\boldsymbol{y}$ is concave in $\boldsymbol{\xi}$
3. sets Ξ and $\mathrm{conv}(\mathcal{Y}(\boldsymbol{x}))$ are convex by definition

□

Enforcing $\boldsymbol{y} \in \mathrm{conv}(\mathcal{Y}(\boldsymbol{x}))$

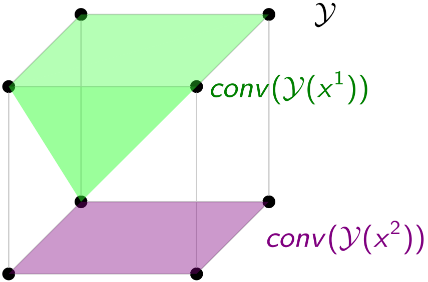
- In general we don't have a compact description for $\mathrm{conv}(\mathcal{Y}(\boldsymbol{x}))$.
- We can write the Dantzig-Wolfe reformulation for $\mathrm{conv}(\mathcal{Y}(\boldsymbol{x}))$ for given $\boldsymbol{x}$.



- We can write a disjunctive formulation! But most probably numerically inconvenient.

Enforcing $\boldsymbol{y} \in \mathrm{conv}(\mathcal{Y}(\boldsymbol{x}))$

- In general we don't have a compact description for $\mathrm{conv}(\mathcal{Y}(\boldsymbol{x}))$.
- We can write the Dantzig-Wolfe reformulation for $\mathrm{conv}(\mathcal{Y}(\boldsymbol{x}))$ for given $\boldsymbol{x}$.

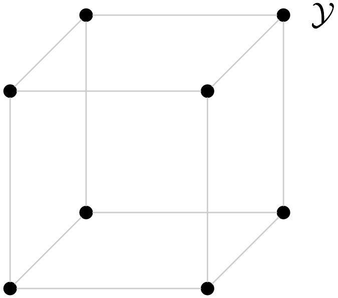


- We can write a disjunctive formulation! But most probably numerically inconvenient.

We identify some conditions where the disjunctive formulation can be avoided, and exploit this more convenient structure.

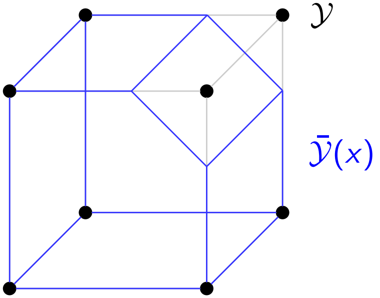
A computationally attractive relaxation

- $\bar{\boldsymbol{y}}^j$ for $j \in \mathcal{L} = \{1, \ldots, L\}$: extreme point solutions of $\mathrm{conv}(\mathcal{Y})$
- $\mathrm{conv}(\mathcal{Y}) := \left\{ \sum_{j \in \mathcal{L}} \bar{\boldsymbol{y}}^j \alpha^j \middle| \boldsymbol{\alpha} \in \Delta^L \right\}$ with $\Delta^L = \left\{ \alpha \in [0,1]^L \middle| \sum_{j=1}^{L} \alpha^j = 1 \right\}$
- $\bar{\mathcal{Y}}(\boldsymbol{x}) = \mathrm{conv}(\mathcal{Y}) \cap \{\boldsymbol{H}\boldsymbol{y}_1 \leq \boldsymbol{d} - \boldsymbol{T}\boldsymbol{x}_1\}$ for $\boldsymbol{x} \in \mathcal{X}$
- $\mathrm{conv}(\mathcal{Y}(\boldsymbol{x})) = \mathrm{conv}\left(\{\boldsymbol{y} \in \mathcal{Y} \mid \boldsymbol{H}\boldsymbol{y}_1 \leq \boldsymbol{d} - \boldsymbol{T}\boldsymbol{x}_1\}\right)$



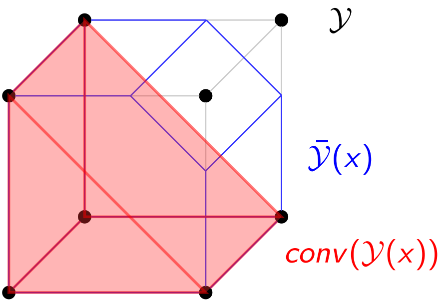
A computationally attractive relaxation

- $\bar{\boldsymbol{y}}^j$ for $j \in \mathcal{L} = \{1, \ldots, L\}$: extreme point solutions of $\mathrm{conv}(\mathcal{Y})$
- $\mathrm{conv}(\mathcal{Y}) := \left\{ \sum_{j \in \mathcal{L}} \bar{\boldsymbol{y}}^j \alpha^j \middle| \boldsymbol{\alpha} \in \Delta^L \right\}$ with $\Delta^L = \left\{ \alpha \in [0,1]^L \middle| \sum_{j=1}^{L} \alpha^j = 1 \right\}$
- $\bar{\mathcal{Y}}(\boldsymbol{x}) = \mathrm{conv}(\mathcal{Y}) \cap \{\boldsymbol{H}\boldsymbol{y}_1 \leq \boldsymbol{d} - \boldsymbol{T}\boldsymbol{x}_1\}$ for $\boldsymbol{x} \in \mathcal{X}$
- $\mathrm{conv}(\mathcal{Y}(\boldsymbol{x})) = \mathrm{conv}\left(\{\boldsymbol{y} \in \mathcal{Y} \mid \boldsymbol{H}\boldsymbol{y}_1 \leq \boldsymbol{d} - \boldsymbol{T}\boldsymbol{x}_1\}\right)$



A computationally attractive relaxation

- $\bar{\boldsymbol{y}}^j$ for $j \in \mathcal{L} = \{1, \ldots, L\}$: extreme point solutions of $\mathrm{conv}(\mathcal{Y})$
- $\mathrm{conv}(\mathcal{Y}) := \left\{ \sum_{j \in \mathcal{L}} \bar{\boldsymbol{y}}^j \alpha^j \middle| \boldsymbol{\alpha} \in \Delta^L \right\}$ with $\Delta^L = \left\{ \alpha \in [0,1]^L \middle| \sum_{j=1}^L \alpha^j = 1 \right\}$
- $\bar{\mathcal{Y}}(\boldsymbol{x}) = \mathrm{conv}(\mathcal{Y}) \cap \{\boldsymbol{H}\boldsymbol{y}_1 \leq \boldsymbol{d} - \boldsymbol{T}\boldsymbol{x}_1\}$ for $\boldsymbol{x} \in \mathcal{X}$
- $\mathrm{conv}(\mathcal{Y}(\boldsymbol{x})) = \mathrm{conv}\left(\{\boldsymbol{y} \in \mathcal{Y} \mid \boldsymbol{H}\boldsymbol{y}_1 \leq \boldsymbol{d} - \boldsymbol{T}\boldsymbol{x}_1\}\right)$



A computationally attractive relaxation

- $\bar{\boldsymbol{y}}^j$ for $j \in \mathcal{L} = \{1, \ldots, L\}$: extreme point solutions of $\mathrm{conv}(\mathcal{Y})$
- $\mathrm{conv}(\mathcal{Y}) := \left\{ \sum_{j \in \mathcal{L}} \bar{\boldsymbol{y}}^j \alpha^j \middle| \boldsymbol{\alpha} \in \Delta^L \right\}$ with $\Delta^L = \left\{ \alpha \in [0,1]^L \middle| \sum_{j=1}^L \alpha^j = 1 \right\}$
- $\bar{\mathcal{Y}}(\boldsymbol{x}) = \mathrm{conv}(\mathcal{Y}) \cap \{\boldsymbol{H}\boldsymbol{y}_1 \leq \boldsymbol{d} - \boldsymbol{T}\boldsymbol{x}_1\}$ for $\boldsymbol{x} \in \mathcal{X}$
- $\mathrm{conv}(\mathcal{Y}(\boldsymbol{x})) = \mathrm{conv}\left(\{\boldsymbol{y} \in \mathcal{Y} \mid \boldsymbol{H}\boldsymbol{y}_1 \leq \boldsymbol{d} - \boldsymbol{T}\boldsymbol{x}_1\}\right)$

$$\begin{aligned} & \min_{\boldsymbol{x} \in \mathcal{X}, \boldsymbol{y} \in \mathrm{conv}(\mathcal{Y}(\boldsymbol{x}))} \boldsymbol{c}^\top \boldsymbol{x} + \max_{\boldsymbol{\xi} \in \Xi} \boldsymbol{\xi}^\top \boldsymbol{Q}\boldsymbol{y} \\ \geq & \min_{\boldsymbol{x} \in \mathcal{X}, \boldsymbol{y} \in \bar{\mathcal{Y}}(\boldsymbol{x})} \boldsymbol{c}^\top \boldsymbol{x} + \max_{\boldsymbol{\xi} \in \Xi} \boldsymbol{\xi}^\top \boldsymbol{Q}\boldsymbol{y} \end{aligned}$$

A computationally attractive relaxation

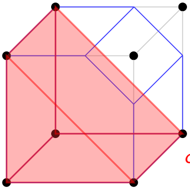
- $\bar{\boldsymbol{y}}^j$ for $j \in \mathcal{L} = \{1, \ldots, L\}$: extreme point solutions of $\operatorname{conv}(\mathcal{Y})$
- $\operatorname{conv}(\mathcal{Y}) := \left\{ \sum_{j \in \mathcal{L}} \bar{\boldsymbol{y}}^j \alpha^j \middle| \boldsymbol{\alpha} \in \Delta^L \right\}$ with $\Delta^L = \left\{ \alpha \in [0,1]^L \middle| \sum_{j=1}^L \alpha^j = 1 \right\}$
- $\bar{\mathcal{Y}}(\boldsymbol{x}) = \operatorname{conv}(\mathcal{Y}) \cap \{\boldsymbol{H}\boldsymbol{y}_1 \leq \boldsymbol{d} - \boldsymbol{T}\boldsymbol{x}_1\}$ for $\boldsymbol{x} \in \mathcal{X}$
- $\operatorname{conv}(\mathcal{Y}(\boldsymbol{x})) = \operatorname{conv}\left(\{\boldsymbol{y} \in \mathcal{Y} \mid \boldsymbol{H}\boldsymbol{y}_1 \leq \boldsymbol{d} - \boldsymbol{T}\boldsymbol{x}_1\}\right)$

$$
\begin{aligned}
& \min_{\boldsymbol{x} \in \mathcal{X}, \boldsymbol{y} \in \operatorname{conv}(\mathcal{Y}(\boldsymbol{x}))} \quad \boldsymbol{c}^\top \boldsymbol{x} + \max_{\boldsymbol{\xi} \in \Xi} \ \boldsymbol{\xi}^\top \boldsymbol{Q}\boldsymbol{y} \\
\geq & \min_{\boldsymbol{x} \in \mathcal{X}, \boldsymbol{y} \in \bar{\mathcal{Y}}(\boldsymbol{x})} \quad \boldsymbol{c}^\top \boldsymbol{x} + \max_{\boldsymbol{\xi} \in \Xi} \ \boldsymbol{\xi}^\top \boldsymbol{Q}\boldsymbol{y} \\
= & \min_{\boldsymbol{x} \in \mathcal{X}, \boldsymbol{\alpha} \in \Delta^L} \quad \boldsymbol{c}^\top \boldsymbol{x} + \max_{\boldsymbol{\xi} \in \Xi} \ \boldsymbol{\xi}^\top \boldsymbol{Q} \sum_{j \in \mathcal{L}} \alpha^j \bar{\boldsymbol{y}}^j \\
& \text{s.t.} \quad \boldsymbol{H} \sum_{j \in \mathcal{L}} \alpha^j \bar{\boldsymbol{y}}_1^j \leq \boldsymbol{d} - \boldsymbol{T}\boldsymbol{x}_1
\end{aligned}
$$

- Now the inner maximization problem can be dualized to obtain a deterministic equivalent formulation.
- We can solve this formulation by column generation.

A sufficient condition for equivalence $\mathrm{conv}(\mathcal{Y}(\boldsymbol{x})) = \bar{\mathcal{Y}}(\boldsymbol{x})$

- $\mathcal{Y}(\boldsymbol{x}) = \{\boldsymbol{y} \in \mathcal{Y} | \boldsymbol{H}\boldsymbol{y}_1 \leq \boldsymbol{d} - \boldsymbol{T}\boldsymbol{x}_1\}$
- $\bar{\mathcal{Y}}(\boldsymbol{x}) = \{\boldsymbol{y} \in \mathrm{conv}(\mathcal{Y}) \mid \boldsymbol{H}\boldsymbol{y}_1 \leq \boldsymbol{d} - \boldsymbol{T}\boldsymbol{x}_1\}$ for $\boldsymbol{x} \in \mathcal{X}$



Proposition

If $H = I$, $T = -I$ and $d = 0$, then $\bar{\mathcal{Y}}(\boldsymbol{x}) = \mathrm{conv}(\mathcal{Y}(\boldsymbol{x}))$.

Possible variations:

- $\boldsymbol{y}_1 \leq \boldsymbol{x}_1$, $\boldsymbol{y}_1 \leq \boldsymbol{1} - \boldsymbol{x}_1$, $\boldsymbol{y}_1 \geq \boldsymbol{x}_1$ (robust knapsack problem with recourse)
- $\boldsymbol{1}^\top \boldsymbol{y}_1 \leq \boldsymbol{x}_1$ (robust single machine scheduling - WIP)

What if $\mathrm{conv}(\mathcal{Y}(\boldsymbol{x})) \neq \bar{\mathcal{Y}}(\boldsymbol{x})$?

1. Add combinatorial Benders' cuts
   - We may add cuts that forbid certain columns based on the first-stage solution
   - These cuts are non-robust cuts (they change col. gen. subproblem structure)
   - Updating the pricing problem is necessary
   - Seems numerically inefficient
2. Lift the recourse feasible region
   - Create a copy of the first-stage decisions in the recourse problem
   - Lets us fall back onto the previous column generation framework

Lifting the recourse feasible region

- Reformulate the recourse feasible region

$$\mathcal{Y}'(\boldsymbol{x}) = \left\{ \boldsymbol{y} \in \mathcal{Y}, \boldsymbol{z} \in \{0,1\}^{L_1} \middle| \boldsymbol{Hy} \leq \boldsymbol{d} - \boldsymbol{Tz}, \boldsymbol{z} \leq \boldsymbol{x}_1, \boldsymbol{z} \geq \boldsymbol{x}_1 \right\}$$

- Let $\mathcal{Y}' = \left\{ \boldsymbol{y} \in \mathcal{Y}, \boldsymbol{z} \in \{0,1\}^{L_1} \middle| \boldsymbol{Hy} \leq \boldsymbol{d} - \boldsymbol{Tz} \right\}$
- $(\bar{\boldsymbol{y}}, \bar{\boldsymbol{z}})^j$ for $j \in \mathcal{L}' = \{1, \ldots, L'\}$: extreme points of $\mathrm{conv}(\mathcal{Y}')$
- Deterministic equivalent formulation:

$$\begin{aligned} \min_{\boldsymbol{x} \in \mathcal{X}, \boldsymbol{\alpha} \in \Delta^{L'}} \quad & \boldsymbol{c}^\top \boldsymbol{x} + \max_{\boldsymbol{\xi} \in \Xi} \ \boldsymbol{\xi}^\top \boldsymbol{Q} \sum_{j \in \mathcal{L}'} \alpha^j \bar{\boldsymbol{y}}^j \\ \text{s.t.} \quad & \boldsymbol{x}_1 = \sum_{j \in \mathcal{L}'} \alpha^j \bar{\boldsymbol{z}}^j \end{aligned}$$

Outline

Two-stage robust capital budgeting (RCB) with loans

Variant of (RCB) introduced in Hanasusanto et al. (2015)

- Investment budget B, which can be extended with loans at first and second stages.
- Set of projects $\mathcal{N} = \{1, \ldots, N\}$.
- Each project has an investment cost c_i and nominal profit $\bar{p}_i$.
- M risk factors, whose values are denoted by $\boldsymbol{\xi} \in \Xi$.
- Actual profit is $\tilde{p}_i(\boldsymbol{\xi}) = (1 + Q_i^\top \boldsymbol{\xi}/2)\bar{p}_i$ for $i \in \mathcal{N}$.
- First-stage decisions:
  (i) take out loan (amount C_1) or not,
  (ii) choose a subset of projects.
- Uncertainty: risks factors $\xi \in \Xi$ are known, revealing $\tilde{p}_i(\xi)$.
- After observing the risk factors, late investments are possible:
  (i) take out loan (amount C_2) or not,
  (ii) choose more projects, which have a degraded profit $f\tilde{p}_i(\xi)$, with $f \in [0, 1)$.

Mathematical formulation

- x_0: the decision to take out an early loan
- x_i: the decision to invest in a project at first stage
- y_0: the decision to take out a late loan
- y_i: the decision to invest in a project at first or second stage
- $\boldsymbol{\xi} \in \Xi = [-1, 1]^M$
- $F_i(\boldsymbol{x}, \boldsymbol{\xi}, \boldsymbol{y})$: net return of project i given $(\boldsymbol{x}, \boldsymbol{\xi}, \boldsymbol{y})$

$$\max_{(\boldsymbol{x},x_0)\in\mathcal{X}} \min_{\boldsymbol{\xi}\in\Xi} \max_{(\boldsymbol{y},y_0)\in\mathcal{Y}(\boldsymbol{x},x_0)} \sum_{i\in\mathcal{N}} F_i(\boldsymbol{x}, \boldsymbol{\xi}, \boldsymbol{y}) - \lambda x_0 - \lambda\mu y_0$$

where

$$F_i(\boldsymbol{x}, \boldsymbol{\xi}, \boldsymbol{y}) = \begin{cases} 0 & \text{if } x_i = y_i = 0 \\ p_i(\boldsymbol{\xi}) & \text{if } x_i = y_i = 1 \\ f\ p_i(\boldsymbol{\xi}) & \text{if } x_i = 0 \text{ and } y_i = 1 \end{cases}$$

Note: F_i can easily be linearized.

Mathematical formulation

- x_0: the decision to take out an early loan
- x_i: the decision to invest in a project at first stage
- y_0: the decision to take out a late loan
- y_i: the decision to invest in a project at first or second stage
- $\boldsymbol{\xi} \in \Xi = [-1, 1]^M$
- $F_i(\boldsymbol{x}, \boldsymbol{\xi}, \boldsymbol{y})$: net return of project i given $(\boldsymbol{x}, \boldsymbol{\xi}, \boldsymbol{y})$

$$\max_{(\boldsymbol{x}, x_0) \in \mathcal{X}} \min_{\boldsymbol{\xi} \in \Xi} \max_{(\boldsymbol{y}, y_0) \in \mathcal{Y}(\boldsymbol{x}, x_0)} \sum_{i \in \mathcal{N}} F_i(\boldsymbol{x}, \boldsymbol{\xi}, \boldsymbol{y}) - \lambda x_0 - \lambda \mu y_0$$

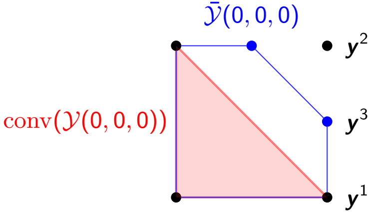
where

$$\mathcal{X} = \left\{ (\boldsymbol{x}, x_0) \in \{0,1\}^{N+1} \middle| \quad \boldsymbol{c}^\top \boldsymbol{x} - C_1 x_0 \leq B \right\}$$

$$\mathcal{Y}(\boldsymbol{x}, x_0) = \left\{ (\boldsymbol{y}, y_0) \in \{0,1\}^{N+1} \middle| \quad \begin{array}{l} \boldsymbol{c}^\top \boldsymbol{y} - C_2 y_0 \leq B + C_1 x_0 \\ y_i \geq x_i \quad \forall i \in \mathcal{N} \end{array} \right\}.$$

Non-tight relaxation: $\bar{\mathcal{Y}}(\boldsymbol{x}, x_0) \neq \mathrm{conv}(\mathcal{Y}(\boldsymbol{x}, x_0))$

$$\mathcal{Y}(\boldsymbol{x}, x_0) = \left\{ \boldsymbol{y} \in \{0,1\}^2 \;\middle|\; \begin{array}{l} 2y_1 + 2y_2 \leq 3 + x_0 \\ y_i \geq x_i \quad \forall i \in \mathcal{N} \end{array} \right\}$$



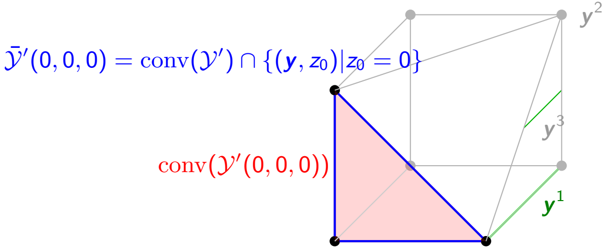
Consider $\boldsymbol{x}^* = (0,0,0)$:

- $\boldsymbol{y}^1 = (1,0)$ is feasible for $\boldsymbol{x}^*$
- $\boldsymbol{y}^2 = (1,1)$ is not feasible for $\boldsymbol{x}^*$
- $\boldsymbol{y}^3 = \frac{1}{2}(\boldsymbol{y}^1 + \boldsymbol{y}^2) \in \bar{\mathcal{Y}}(0,0,0)$ **but** $\boldsymbol{y}^3 \notin \mathrm{conv}(\mathcal{Y}(0,0,0))$

Second-stage fractional solution $\boldsymbol{y}^3$ can only be made from the infeasible solution $\boldsymbol{y}^2$

An alternative formulation through lifting

$$\mathcal{Y}'(\boldsymbol{x}, x_0) = \left\{ (\boldsymbol{y}, z) \in \{0,1\}^3 \;\middle|\; \begin{array}{l} 2y_1 + 2y_2 \leq 3 + z_0 \\ y_i \geq x_i \quad \forall i \in \mathcal{N} \\ z_0 = x_0 \end{array} \right\}$$



Consider $\boldsymbol{x}^* = (0,0,0)$:

- $\boldsymbol{y}^1 = (1, 0, \rho)$ is feasible for $\boldsymbol{x}^*$
- $\boldsymbol{y}^2 = (1, 1, 1) \in \mathcal{Y}'$
- $\boldsymbol{y}^3 = \frac{1}{2}(\boldsymbol{y}^1 + \boldsymbol{y}^2) = (1, 0.5, 0.5 + \rho/2) \notin \bar{\mathcal{Y}}'(x^*)$

An alternative formulation through lifting

- Add a copy of x_0 to the recourse feasible region

$$\mathcal{Y}'(\boldsymbol{x}, x_0) = \left\{ (\boldsymbol{y}, y_0, z_0) \in \{0,1\}^{N+2} \;\middle|\; \begin{array}{l} \boldsymbol{c}^\top \boldsymbol{y} \leq B + C_1 z_0 + C_2 y_0 \\ y_i \geq x_i \quad \forall \boldsymbol{i} \in \mathcal{N} \\ z_0 = x_0 \end{array} \right\}.$$

- The constraint $c^\top \boldsymbol{y} \leq B + C_1 z_0 + C_2 y_0$ is now purely recourse.

Column generation subproblem

Let us recall our deterministic equivalent formulation:

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathcal{X},\boldsymbol{\alpha}\in\Delta^{L'}} \quad & \boldsymbol{c}^\top \boldsymbol{x} + \max_{\boldsymbol{\xi}\in\Xi} \boldsymbol{\xi}^\top \boldsymbol{Q} \sum_{j\in\mathcal{L}'} \alpha^j \bar{\boldsymbol{y}}^j \\ \text{s.t.} \quad & \boldsymbol{x}_1 = \sum_{j\in\mathcal{L}'} \alpha^j \bar{\boldsymbol{z}}^j \end{aligned}$$

The pricing problem takes the form,

$$-\kappa + \max_{(\boldsymbol{y},y_0,z_0)\in\mathcal{Y}'} -\lambda\mu y_0 + \tilde{c}_{z0} z_0 + \sum_{i\in\mathcal{N}} \tilde{c}_{yi} y_i$$

with $\mathcal{Y}' = \left\{(\boldsymbol{y}, y_0, z_0) \in \{0,1\}^{N+2} \mid \boldsymbol{c}^\top \boldsymbol{y} \leq B + C_1 z_0 + C_2 y_0\right\}$ and $\tilde{c}$ the reduced costs, which is a classical knapsack problem (by substituting $\bar{y}_0 = 1 - y_0$ and $\bar{z}_0 = 1 - z_0$).

Remark

The subproblem can be solved through a dynamic programming algorithm.

Outline

Instance Generation

- Instances inspired by those presented in Hanasusanto et al. (2015).
- For given number of items N, and parameters $R = 100$ and $H = 100$, $h \in \{20, 40, 60, 80\}$
- Costs c_i for $i \in \mathcal{N}$: uniformly from the interval $[1, R]$
- Nominal profits: $\bar{\boldsymbol{p}} = \boldsymbol{c}/5$
- Investment budget: $B = \frac{h}{H+1} \sum_{i \in \mathcal{I}} c_i$
- Postponed investments generate %80 of the profits, that is $f = 0.8$.
- The loans C_1 and C_2: %20 of the initial budget B
- Interest values: $\lambda = \frac{0.12}{5}$ and $\mu = 1.2$.

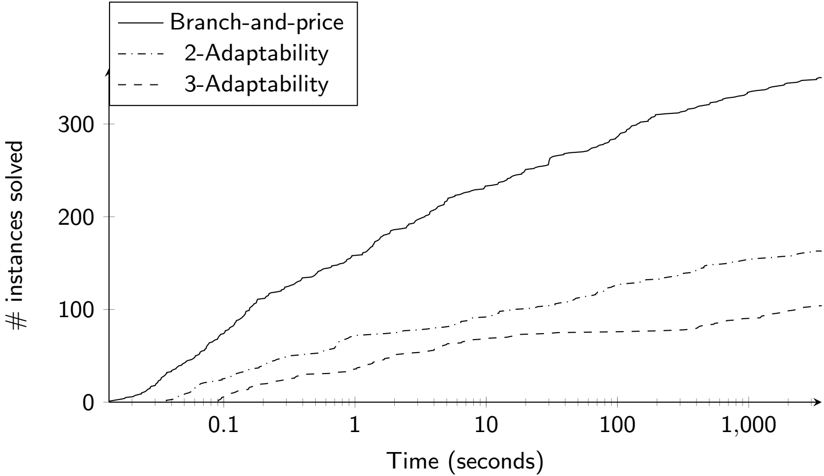
Experimental setting

We compare the performance of a branch-and-price algorithm solving our lifted-recourse reformulation with solving 2- and 3-adaptability models with a commercial solver.

- Generic Branch-and-Price algorithm implemented in C++ (BaPCod library)
  - Stabilization of column generation by smoothing of dual variables
  - Choice of first-stage variables to branch on with strong branching
  - Primal diving heuristic
  - Single-threaded
  - Column generation subproblems are solved by straightforward array-based label-correcting algorithm
  - No other problem-specific development
- (MI)LP models are solved using IMB ILOG Cplex 12.9 through C++ Concert Library with default settings (4 threads)

All tests are run on a 4-core Linux machine equipped with 20 GB RAM.
The time limit is 1 hour for each run.

Solution time and scalability



Number of instances solved by each method through time.
60 instances for each $N \in \{10, 20, 30, 40, 50, 100\}$.
Our B&P is 2 to 4 orders of magnitude faster than MILP solver on 2- and 3-adapt models

Solution time and scalability

Number of instances solved to convergence by each method within one hour.

	$N = 10$	$N = 20$	$N = 30$	$N = 40$	$N = 50$	$N = 100$
B&P	60	60	58	55	60	57
2-Adapt	60	60	21	15	7	0
3-Adapt	60	31	13	0	0	0

- K-adaptable models are hard to solve because of their poor linear relaxation
- Our B&P scales well, but fails at solving some relatively small-size instances
  Typical relative gap after a few minutes: $10^{-}3$ *or less.*
  Probably suffers from many good infeasible solutions of the first-stage problem linear relaxation (add cuts?).

Full adjustability VS K-adaptability

Number of instances categorized by the number of active columns in the best primal solution reported by the branch-and-price algorithm.

	$N = 10$	$N = 20$	$N = 30$	$N = 40$	$N = 50$	$N = 100$
K=1	33	17	16	14	10	10
K=2	18	19	3	4	6	5
K=3	5	8	6	4	0	0
K=4	2	11	13	3	5	1
K=[5,9]	2	5	22	35	39	44

- A number of instances admit robust static optimal solutions
- For larger instances, 2- and 3-adaptable solutions are (most probably) not optimal

Full adjustability VS K-adaptability

Only instances with at least 3 columns in optimal solutions are considered.

Gaps between the best primal solutions provided by the K-adaptability method and the branch-and-price algorithm.

	$N = 10$	$N = 20$	$N = 30$	$N = 40$	$N = 50$	$N = 100$
Avg 2-Adapt (%)	1.91	0.27	0.38	0.44	0.32	0.12
Max 2-Adapt (%)	6.29	1.36	1.50	2.01	1.20	0.50
Avg 3-Adapt (%)	0.28	0.19	0.61	0.44	0.35	0.14
Max 3-Adapt (%)	0.99	1.97	3.57	1.43	1.09	0.75

$$Gap = 2\frac{\text{Best K-adapt solution} - \text{Best B\&P solution}}{\text{Best B\&P solution} + \text{Best B\&P solution}}$$

K-adaptable models turn out to be very good approximations of the fully adjustable problem

Outline

Conclusions and future work

1. Conclusions:
   - We study a special class of two-stage binary robust optimization problems with objective uncertainty.
   - Single-stage relaxation that can be solved using column generation approaches.
   - Special case where the relaxation is tight, and a technique to reduce to this case.
   - Application to a version of the two-stage robust capital budgeting problem.
   - Computational results that show the effectiveness of the column generation approach.
2. Future Work:
   - Extend the computational results to different problems.
     Done: Similar computational results on a two-stage robust knapsack problem.
   - Develop other techniques for non-tight relaxation.
3. Room for problem-specific improvements:
   - Master program/First-stage formulation reinforcement (KSP cover cuts, dominances. . .)
   - Efficient solver for column generation pricing problems. . .
   - Generate batches of columns instead of a single one. . .

Available on Optimization Online:
"Decomposition-based approaches for a class of two-stage robust binary optimization problems"

Disjunctive MIP reformulation

- $\boldsymbol{x}^i$ for $i \in \mathcal{K} = \{1, \ldots, K\}$: extreme point solutions of $\mathrm{proj}_{\{0,1\}^{N_1}} \mathcal{X}$
- $\bar{\boldsymbol{y}}^j$ for $j \in \mathcal{L} = \{1, \ldots, L\}$: extreme point solutions of $\mathcal{Y}$
- $\mathcal{L}_i = \{j \in \mathcal{L} \mid \boldsymbol{H}\bar{\boldsymbol{y}}_1^j \leq \boldsymbol{d} - \boldsymbol{T}\boldsymbol{x}_1^i\}$ for $i = 1, \ldots, K$.
- $\Delta^n = \left\{\alpha \in [0,1]^n \middle| \sum_{j=1}^n \alpha^j = 1\right\}$, for $n \in \mathbb{N}$: n-dimensional simplex

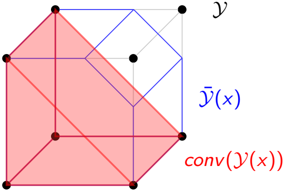
Proposition

$\mathrm{conv}(\mathcal{Y}(\boldsymbol{x}^i)) = \left\{\sum_{j \in \mathcal{L}_i} \alpha^j \bar{\boldsymbol{y}}^j \middle| \boldsymbol{\alpha} \in \Delta^{|\mathcal{L}_i|}\right\}$ *for* $i \in \mathcal{K}$.

$$\begin{aligned}
\min \quad & c^T \boldsymbol{x} + \max_{\boldsymbol{\xi} \in \Xi} \ \boldsymbol{\xi}^\top \boldsymbol{Q} \sum_{i \in \mathcal{K}} \sum_{j \in \mathcal{L}_i} \alpha_i^j \bar{\boldsymbol{y}}^j \\
\text{s.t.} \quad & \sum_{j \in \mathcal{L}_i} \alpha_i^j \leq \mathbb{I}_{\boldsymbol{x}^i} \quad \forall i \in \mathcal{K} \\
& \sum_{i \in \mathcal{K}} \sum_{j \in \mathcal{L}_i} \alpha_i^j = 1 \\
& \mathbb{I}_{\boldsymbol{x}^i} = 1 \Leftrightarrow \boldsymbol{x} = \boldsymbol{x}^i \quad \forall i \in \mathcal{K} \\
& \mathbb{I}_{\boldsymbol{x}^i} \in \{0,1\} \quad \forall i \in \mathcal{K}, \boldsymbol{\alpha}_i \in [0,1]^{|\mathcal{L}_i|} \quad \forall i \in \mathcal{K}, \boldsymbol{x} \in \mathcal{X}
\end{aligned}$$

A sufficient condition for equivalence $\mathrm{conv}(\mathcal{Y}(\boldsymbol{x})) = \bar{\mathcal{Y}}(\boldsymbol{x})$

- $\mathcal{Y}(\boldsymbol{x}) = \{\boldsymbol{y} \in \mathcal{Y} | \boldsymbol{H}\boldsymbol{y}_1 \leq \boldsymbol{d} - \boldsymbol{T}\boldsymbol{x}_1, \boldsymbol{A}\boldsymbol{y} \leq \boldsymbol{b}\}$
- $\bar{\mathcal{Y}}(\boldsymbol{x}) = \mathrm{conv}(\mathcal{Y}) \cap \{\boldsymbol{y} | \boldsymbol{H}\boldsymbol{y}_1 \leq \boldsymbol{d} - \boldsymbol{T}\boldsymbol{x}_1\}$ for $\boldsymbol{x} \in \mathcal{X}$



Proposition

If $H = I$, $T = -I$ and $d = 0$, then $\bar{\mathcal{Y}}(\boldsymbol{x}) = \mathrm{conv}(\mathcal{Y}(\boldsymbol{x}))$.

Proof.

Under the given assumptions, $\mathcal{Y}(\boldsymbol{x}) = \{\boldsymbol{y} \in \mathcal{Y} | \boldsymbol{y}_1 \leq \boldsymbol{x}_1, \boldsymbol{A}\boldsymbol{y} \leq \boldsymbol{b}\}$.
We already have that $\mathrm{conv}(\mathcal{Y}(\boldsymbol{x})) \subseteq \bar{\mathcal{Y}}(\boldsymbol{x})$.
Assume $\boldsymbol{y} = \sum_{j \in \mathcal{J}} \bar{\boldsymbol{y}}^j \alpha^j \in \bar{\mathcal{Y}}(\boldsymbol{x})$ and $\boldsymbol{y} \notin \mathrm{conv}(\mathcal{Y}(\boldsymbol{x}))$.
Then $\sum_{j \in \mathcal{J}} \alpha^j = 1$, and $\sum_{j | \boldsymbol{y}_1^j \leq \boldsymbol{x}_1} \alpha^j < 1$.
So there is a linking constraint i and $k \in \mathcal{J}$ such that $\bar{y}_i^k > x_i$ and $\alpha_k > 0$.
Since $x_i \in \{0,1\}$ and $\bar{y}_i^k \in \{0,1\}$, $x_i = 0$ and $\bar{y}_i^k = 1$. This implies

$$y_i = \sum_{j \in \mathcal{J}} \bar{y}_i^j \alpha^j \geq \alpha_k \bar{y}_i^k = \alpha_k > 0 = x_i$$

That contradicts $\boldsymbol{y} \in \bar{\mathcal{Y}}(\boldsymbol{x}) \rightarrow \bar{\mathcal{Y}}(\boldsymbol{x}) \subseteq \mathrm{conv}(\mathcal{Y}(\boldsymbol{x}))$. □

Reformulation through enumeration

- X^0: set of first-stage feasible solutions with cardinality $|X^0|$
- $\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{r} \end{bmatrix}^j \in \mathcal{Y}(\boldsymbol{x}^i)$: set of second-stage feasible solutions corresponding to $\boldsymbol{x}^i$ for $i = 1, \ldots, |X^0|$

$$\min_{x \in \mathcal{X}} \max_{\xi \in \Xi} \quad \theta$$

$$\text{s.t.} \quad \theta \leq \boldsymbol{c}^\top \boldsymbol{x} + \boldsymbol{q}(\boldsymbol{\xi})^\top \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{r} \end{bmatrix}^j \quad \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{r} \end{bmatrix}^j \in \mathcal{Y}(\boldsymbol{x})$$

$$\begin{aligned} \max \quad & \theta^i \\ \text{s.t.} \quad & \theta^i \leq \boldsymbol{c}^\top \boldsymbol{x}^i + \boldsymbol{q}(\boldsymbol{\xi}^i)^\top \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{r} \end{bmatrix}^j \quad \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{r} \end{bmatrix}^j \in \mathcal{Y}(\boldsymbol{x}^i) \\ & \boldsymbol{\xi}^i \in \Xi \end{aligned}$$

$$\begin{aligned} \max \quad & \Theta \\ \text{s.t.} \quad & \Theta \leq \theta^i \quad \forall i = 1, \ldots, |X^0| \\ & \theta^i \leq \boldsymbol{c}^\top \boldsymbol{x}^i + \boldsymbol{q}(\boldsymbol{\xi}^i)^\top \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{r} \end{bmatrix}^j \quad \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{r} \end{bmatrix}^j \in \mathcal{Y}(\boldsymbol{x}^i) \\ & \boldsymbol{\xi}^i \in \Xi \quad \forall i = 1, \ldots, |X^0| \end{aligned}$$

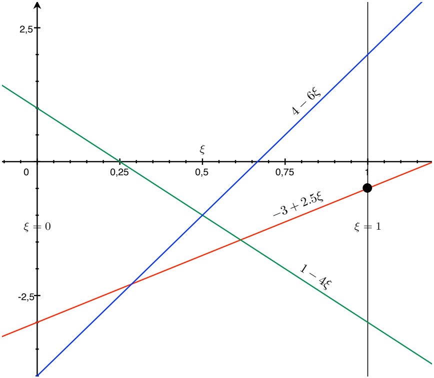
- $c^\top \boldsymbol{x}^i = \sum_{k \in \mathcal{I}} (f_k - p_k) x_k^i$
- $q(\boldsymbol{\xi}^i)^\top \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{r} \end{bmatrix}^j = \sum_{k \in \mathcal{I}} (\bar{d}_k \xi_k^i - f_k) y_k^j - \bar{d}_k \xi_k^i r_k^j$

An instructive example

- Consider the static robust optimization problem

$$\min_{x\in\{0,1\},\mathbf{y}\in\{0,1\}^3} \max_{\xi\in[0,1]} \quad (-3+2.5\xi)y_1 + (1-4\xi)y_2 + (-4+6\xi)y_3$$

$$\text{s.t.} \quad y_1 + y_2 + y_3 \leq x$$

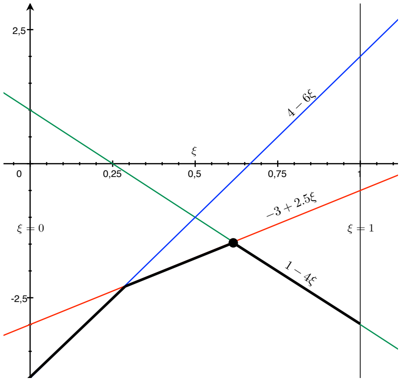


- Optimal solution is $x = y_1 = 1$, $y_2 = y_3 = 0$, with objective value -0.5.

An instructive example

- Adjustable robust optimization problem: Consider $\boldsymbol{y}$ to be wait-and-see decisions.

$$\max_{x\in\{0,1\}} \max_{\xi\in[0,1]} \min_{\boldsymbol{y}\in\{0,1\}^3} \quad (-3+2.5\xi)y_1 + (1-4\xi)y_2 + (-4+6\xi)y_3$$
$$\text{s.t.} \quad y_1 + y_2 + y_3 \leq x$$



- Optimal solution is $x=1$ with value -1.46, convex combination of $y_1=1$ and $y_2=1$.